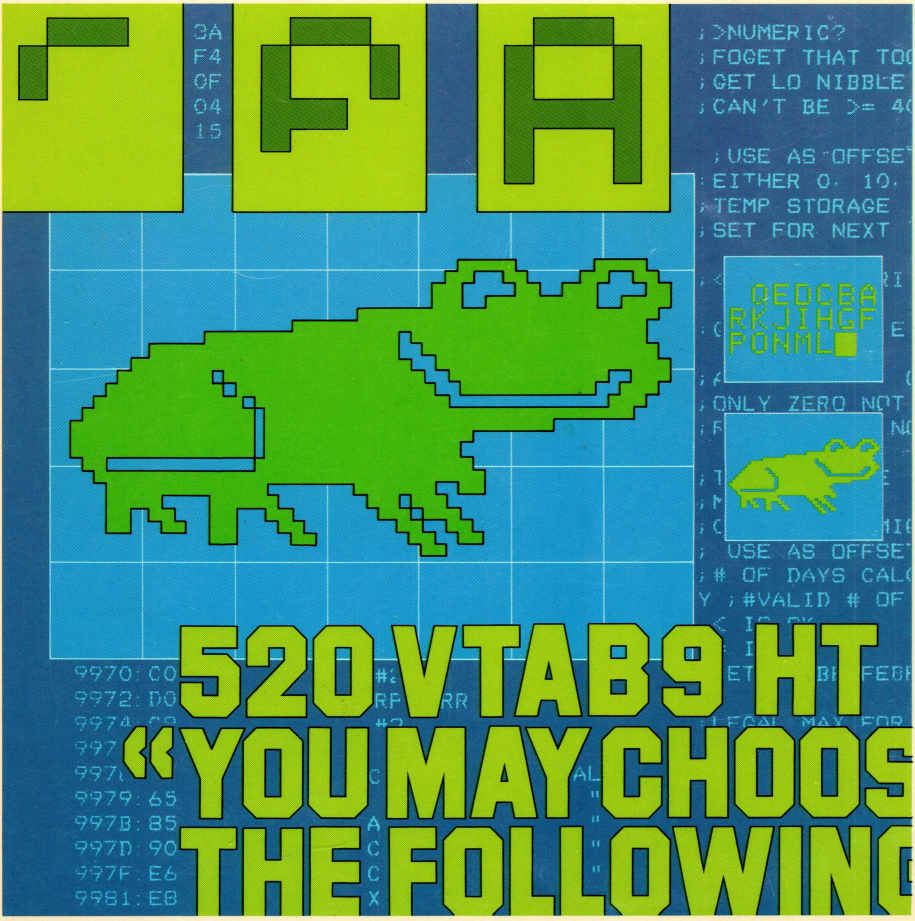
Apple II

Applesoft Tool Kit



NOTICE

Apple Computer Inc. reserves the right to make improvements in the product described in this manual at any time and without notice.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

APPLE COMPUTER INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. APPLE COMPUTER INC. SOFTWARE IS SOLD OR LICENSED "AS IS". THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT APPLE COMPUTER INC., ITS DISTRIBUTOR, OR ITS RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL APPLE COMPUTER INC. BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF APPLE COMPUTER INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

This manual is copyrighted. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Apple Computer Inc.

©198Ø by APPLE COMPUTER INC.
1Ø26Ø Bandley Drive
Cupertino, California 95Ø14
(4Ø8) 996-1Ø1Ø

The word APPLE and the Apple logo are registered trademarks of APPLE COMPUTER INC.

APPLE Product #A2LØØ38
(Ø3Ø-Ø113-ØØ)

Apple II

Applesoft Tool Kit

TABLE OF CONTENTS

INTRODUCTION 1

CHAPTER 1

THE APPLESOFT PROGRAMMER'S ASSISTANT 3

3 Starting Up
3 Using the APA
4 The Commands
4 &Auto
5 &MAnual
5 &Renumber
7 &Hold
7 &Merge
9 &Compress
9 &Length
9 &Show
9 &Noshow
9 &Keys
1Ø &Xref
11 Using the APA Under Program Control

CHAPTER 2

HI-RES CHARACTER GENERATOR 13

14 Loading Character Sets
15 The Commands
2Ø Differences from Standard Text Display
2Ø Leaving the Hi-Res Character Generator
2Ø Using the HRCG Under Program Control

CHAPTER 3

ANIMATRIX 23

23 Starting Up
24 Getting Characters onto the Screen
25 Editing Characters
25 Duplicating Characters
26 Graphic Character Sets
27 Leaving Animatrix
28 Summary of Animatrix Commands

CHAPTER 4

THE RELOCATING LOADER 29

29 Restrictions
29 RBOOT
29 RLOAD

INDEX 31

INTRODUCTION

The Applesoft Tool Kit is designed to help you use the capabilities of the Apple II. The Applesoft Programmer's Assistant gives you the same facilities in Applesoft as you have in Integer BASIC: merging, renumbering, and automatic line numbering during program entry. Other programs make it easier to produce reliable and interesting interactive programs, such as games and educational programs. Animatrix and the High-Resolution Character Generator simplify graphics: the former lets the user define graphic character sets, and the latter makes it easier to display them. The Relocating Loader lets the user write programs that can run on Apples with various amounts of memory, by making assembly-language subroutines relocatable.

The Applesoft Tool Kit also includes three programs, RIBBIT, SKYLAB, and MAXWELL, which demonstrate the use of the High Resolution Character Generator and the Relocating Loader.

The Applesoft Tool Kit is intended for experienced users who are familiar with Applesoft II, DOS 3.3, and in some cases, 65Ø2 assembly language. It is not mainly intended for the novice user, although some programs can be used by novices--the games can be played by children. The DOS Manual and the Applesoft Tutorial or Reference Manual should be read before using the Applesoft Tool Kit.

The Applesoft Tool Kit is designed to use DOS 3.3. It will not run under previous versions of DOS. If you have Applesoft ROMs in your Apple (that is, if you have an Apple II Plus or an Apple II with the Applesoft Firmware Card), you can boot DOS 3.3 from the Tool Kit diskette and run its programs by simply inserting it into drive 1 and following the normal boot procedure. If you have an Apple II with the Language System, you must first boot the DOS 3.3 System Master, to load Applesoft into the Language Card.

The programs described in this manual will run on an Apple II Plus or an Apple II with Applesoft Card or Language System, and at least one Disk II drive and controller. The 16-sector disk controller PROMs, P5A and P6A (or 341-ØØ27-xx and 341-ØØ28-xx, where xx is the batch number), must be installed in each disk controller.

The Hi-Res Character Generator requires either 32K or 48K of memory: 32K if only the first hi-res graphics page is used, 48K if the second page is used. At least 32K of memory is required for the Assembler/Editor and the Applesoft Programmer's Assistant; 48K is required for the other programs on the Tool Kit diskette.

The Assembler/Editor (described in its own manual) does not require Applesoft, and can run on an Apple II without Applesoft Card or Language System, if the Apple has one or more disk drives, with PROMs P5A and P6A in each disk controller, and at least 32K of memory.

CHAPTER 1
THE APPLESOFT PROGRAMMER'S ASSISTANT

The Applesoft Programmer's Assistant (APA) does several things to help you write and change Applesoft programs. It numbers lines automatically as you enter a program, renumbers and merges programs at will, measures the length of a program, and deletes remarks from a program in its final form. It also shows you the control characters in a file, lets you type special characters from the keyboard, and cross-references programs.

STARTING UP

To use this program, make sure DOS is running, and that the Applesoft Tool Kit Diskette is in drive 1, then type

RUN LOADAPA

When you press RETURN, the drive will whir and zick, and the LOADAPA program will relocate, load, and initialize the APA. If you want, you can now LIST the program, to see how it works.

USING THE APA

Once you have examined the LOADAPA program, RUN it. LOADAPA will LOAD APA just below HIMEM, then reset HIMEM just below APA, thereby hiding APA so it can't get erased accidentally. After the APA is loaded and hidden, you will see a title, a copyright message, and a list of commands. These are shown below in a slightly different form from what you'll see on the screen, in order to clarify their syntax. Each of the items in angle brackets, like <start>, is an integer parameter; anything in square brackets, like [, <last>], is optional. Optional items may be nested: you can omit the commas and parameters inside each set of brackets, matching the brackets from the innermost outward. Neither angle brackets nor square brackets should be typed at the keyboard--only the ampersand, the uppercase letter or letters of the command name, the parameters, and the commas between the parameters. The spaces in the syntax diagrams are optional, and are the only ones allowed--don't space after the ampersand.

&Renumber [<start>] [, [<inc>] [, [<first>] [, <last>]]]

&Hold

&Merge

&Length

&Compress

&Show

&Noshow

&Auto <start>[, <inc>]

&MAnual

&Keys

&Xref

These commands are discussed in detail below. The lower-case letters in the command name may be omitted, as may the numeric parameters in square brackets: e.g., [<inc>]. These parameters (following &Renumber and &Auto) are integers between 1 and 63999, and are identified by position: if you leave one out, put in the comma following it so the APA will know it was left out.

THE COMMANDS

&Auto <start>[, <inc>]

This command gives you automatic line numbering. You can use it while entering a program, as the APA "hides" at the top of memory, where it can not clobber, or be clobbered by, your program. If your program is extremely large, you may need the space taken up by the APA, but this will rarely be a problem.

To write a program starting with line number 1ØØ, simply type
&A 1ØØ
and press RETURN. Now press the space bar, and a line number will appear. Type your statement, press RETURN, and press the space bar: the next line number, 11Ø, will appear. Unless you state otherwise, the increment <inc> will be 1Ø.

If you wish to leave a line number unused, press RETURN, then the space bar: the next line number will appear, and no statement will be entered under the previous one.

If you change your mind about a statement while you are typing it, press CTRL-X: the same number will reappear, and no statement will be entered.

Since a line number appears only if you press the space bar after the prompt ( ]), you can type a run-time command any time you see the prompt, as long as you don't begin it with a space. This means you can RUN or LIST your program, or SAVE it and LOAD another, just as if the APA weren't there.

Note: It is possible to clobber the APA with run-time commands. Using the FP command will do it, as will changing HIMEM or the input/output hooks. For this reason, when the APA is loaded, avoid RUNning any program that changes HIMEM or the I/O hooks, or pressing RESET.

&MAnual

To turn off automatic line numbering type
&MA
and press RETURN. &MAnual is the default when you load the APA: you have to ask for automatic line numbering to get it.

&Renumber [<start>] [, [<inc>] [, [<first>] [, <last>]]]

The &Renumber command will renumber any part of your program, with any starting line number and increment, as long as this renumbering will not cause interleaving of lines or duplication of line numbers. Interleaving or duplication would occur if a line range were renumbered so that its new numbers overlapped the numbers of an existing line range. If an &R command would cause this to happen, the command will not be executed.

Let's start with an example. Enter this simple program at the keyboard, and SAVE it as, say, SAMPLE:

```
1Ø PRINT "S"
2Ø PRINT "A"
3Ø PRINT "M"
4Ø PRINT "P"
5Ø PRINT "L"
6Ø PRINT "E"
```

To renumber this program, type
&R
Your whole program will be renumbered, starting with number 1ØØ and incrementing by 1Ø.

```
1ØØ PRINT "S"
11Ø PRINT "A"
12Ø PRINT "M"
13Ø PRINT "P"
14Ø PRINT "L"
15Ø PRINT "E"
```

You can specify a different starting line number <start> and increment <inc> by typing, say,
&R 1ØØØ, 5Ø

where the starting number is 1ØØØ and the increment is 5Ø. This will give you

```
1ØØØ PRINT "S"
1Ø5Ø PRINT "A"
11ØØ PRINT "M"
115Ø PRINT "P"
12ØØ PRINT "L"
125Ø PRINT "E"
```

If you wish to renumber only part of your program, you can specify the <first> and <last> old line numbers to be changed. For example, if you reLOAD SAMPLE, then type

```
&R 1ØØ, 1Ø, 3Ø, 4Ø
```

--where the starting number is 1ØØ, the increment is 1Ø, and the first and last old line numbers are 3Ø and 4Ø, respectively--you will get

```
1Ø PRINT "S"
2Ø PRINT "A"
5Ø PRINT "L"
6Ø PRINT "E"
1ØØ PRINT "M"
11Ø PRINT "P"
```

Only the lines previously numbered 3Ø and 4Ø have been renumbered. Notice that the lines have been reordered to keep the line numbers in order. This feature can be used for such purposes as moving subroutines to the end of your program.

The command above is equivalent to

```
&R , , 3Ø, 4Ø
```

The APA uses the default value whenever it sees a comma without a parameter before it. The default values are

```
<start>     1ØØ
<inc>        1Ø
<first>       Ø
<last>    63999
```

The Renumber command will not let you use parameters that would give two lines the same number or interleave two sets of lines. In this situation, you will get an error message. For example, LOAD SAMPLE and type

```
&R 1Ø, 5, 4Ø
```

You will get the message

```
INTERLEAVED OR DUPLICATE LINE NUMBER
```

because this command would not only put line 5Ø between lines 1Ø and 2Ø but also put lines 4Ø and 6Ø on top of lines 1Ø and 2Ø. In this respect, the APA differs from certain other renumbering programs, which would merrily scramble your file when given such a command.

The Renumber command will change references from one line to another, such as GOTOs, GOSUBs, ONERRs, and the like. So if the program ran correctly before renumbering, it will run correctly after, unless you

changed the order of the lines. Renumber will not change line references in REM statements, so check these yourself.

&Hold

This command hides your program above HIMEM so it can't be erased, and lets you LOAD another program into the area your program just left. To use it, make sure you have a program in memory, then type

```
&H
```

Now try to LIST your program. It will be gone. To bring it back, read on.

&Merge

To get your program out of the Hold area, type

```
&M
```

LIST it, and it will reappear.

You can use this command to merge two programs. Let's start with our SAMPLE program. First LOAD and LIST it on the screen:

```
1Ø PRINT "S"
2Ø PRINT "A"
3Ø PRINT "M"
4Ø PRINT "P"
5Ø PRINT "L"
6Ø PRINT "E"
```

Now delete lines 1Ø, 3Ø, and 5Ø by typing the line numbers, followed by RETURNs. When you LIST the program, you will have

```
2Ø PRINT "A"
4Ø PRINT "P"
6Ø PRINT "E"
```

SAVE this as APE, or whatever. Now type

```
&H
```

and these lines will be hidden.

Now reLOAD SAMPLE, and delete lines 2Ø, 4Ø, and 6Ø, as above. This will LIST as

```
1Ø PRINT "S"
3Ø PRINT "M"
5Ø PRINT "L"
```

SAVE this as SML, or whatever. Now type

```
&M
```

and LIST the result:

```
1Ø PRINT "S"
2Ø PRINT "A"
3Ø PRINT "M"
4Ø PRINT "P"
5Ø PRINT "L"
6Ø PRINT "E"
```

Presto! The program you carved up so diligently is now whole again, and sorted by line number. If you had renumbered SML before merging, for example, by typing

```
&R , 2ØØ
```

so that SML looked like this

```
1ØØ PRINT "S"
3ØØ PRINT "M"
5ØØ PRINT "L"
```

you will get different results. Type

```
&M
```

and you will get

```
2Ø PRINT "A"
4Ø PRINT "P"
6Ø PRINT "E"
1ØØ PRINT "S"
3ØØ PRINT "M"
5ØØ PRINT "L"
```

This command, unlike the Renumber command, can create duplicate line numbers. For example, put APE on Hold, then LOAD SML. Renumber it

```
&R 2Ø, 1Ø
```

and you will have, on LISTing,

```
2Ø PRINT "S"
3Ø PRINT "M"
4Ø PRINT "L"
```

Notice that both APE and the new SML have lines numbered 2Ø and 4Ø. Now type

```
&M
```

and you will get this message

```
DUPLICATE LINE NUMBER
2Ø
4Ø

CONTINUE?
```

If you do, whenever two lines have the same number, the line from the Hold area will be chosen, and the other line with the same number will be deleted. You will now have

```
2Ø PRINT "A"
3Ø PRINT "M"
4Ø PRINT "P"
6Ø PRINT "E"
```

which may be what you had in mind, or may not.

With this command, you can perform wonders, and save yourself much time. You can also wreak considerable havoc, so it is advisable to SAVE each of the programs to be merged, as "all the king's horses and all the king's men ..."

&Compress

This command will remove REMarks from the program in memory. The Compress command will let you maintain two versions of a program: one documented and one compressed. The first will be easier to maintain, and the second will be smaller and run faster. To use this feature, make sure you have SAVEd the current version of your documented program, then type

```
&C
```

The program in memory will be compressed, and can then be SAVEd separately. If you later revise the program, you can change the documented version and make a new compressed version from it.

&Length

This command gives the Length of the program in memory. To use it, type

```
&L
```

and you will get the program's length in bytes, in both decimal and hexadecimal form.

&Show

This command makes the control characters in your program visible. Type

```
&S
```

and control characters will be displayed in inverse video.

&Noshow

This command makes control characters invisible. Type

```
&N
```

and control characters will not be displayed.

&Keys

This command lets you type three additional characters at the keyboard: the underscore, _ ; the backslash, \ ; and the left bracket, [. To use this command, type

```
&K
```

From now on, you can input these characters by typing the following control characters:

Underscore	_	CTRL-O
Backslash	\	CTRL-L
Left bracket	[	CTRL-K

To cancel the &Keys feature after you no longer need it, type
&MA

Note: You can't use both this feature and the &Auto feature at the same time. If you type &K when &Auto is on, you will turn &Auto off and &Keys on; if you type &A when &Keys is on, you will turn &Keys off and &Auto on.

&Xref

The &Xref command gives you a cross-reference listing of an Applesoft program, listing each variable, followed by the numbers of the lines in which it appears. To use it, LOAD the program you wish to cross-reference, then type
&X
and the variables and their line numbers will be listed on the screen. If your cross-reference listing is long, you can interrupt it by typing
CTRL-S
if you have the Autostart ROM in your Apple.

To see how this command works, type
FP
and RUN LOADAPA, then type
&X

You will get this table on the screen.

```
        AD 13Ø, 16Ø, 18Ø
```

showing all the lines where AD, the LOADAPA program's only variable appears. To see how &Xref works on a large program, LOAD the program RIBBIT. When the program is in memory, LIST it and watch its many lines go by on the screen. Now type
&X

You will get a very long cross-reference table, which can be interrupted by the CTRL-S command. Note that all variable names will be shortened to two characters if they are longer than two, as Applesoft only distinguishes the first two characters of a variable name. In the case of a string, these will be followed by a dollar sign, $; in the case of an array, an open parenthesis, (; in the case of a string array, both, $(. Names of integer variables will be followed by a percent sign, % ; integer arrays will be followed by a percent sign and an open parenthesis, %(. For example, the cross-reference table for RIBBIT begins with the variables

```
        A$      (which is a string variable),
        AD      (which is a real variable),
        BC      (which is a real variable),
        BD      (which is a real variable),
        BS      (which is a real variable),
        BU$(    (which is a string array variable),
```

each variable name followed by a set of line numbers.

USING THE APA UNDER PROGRAM CONTROL

You can also RUN the APA program from inside your own program. To do this, copy LOADAPA into your program, using the Hold and Merge commands, and the Renumber command if necessary. You want to be sure, when you do this, that LOADAPA does not interleave with a section of your program.

To see what LOADAPA will do, LOAD LOADAPA without RUNning it, then LIST it on the Apple screen.

If you don't want your program to print the title, change line 16Ø to 16Ø ADRS = USR (Ø) + 3

This will cause the program to start at its second entry point, which skips the title. You can get rid of the other displays as well, by deleting the PRINT statements, except 14Ø PRINT CHR$ (4),"BLOAD RBOOT" which does part of the relocating, and 29Ø PRINT "UNABLE TO LOAD" which is an error message. If line 14Ø is deleted, the Relocating Loader will not be loaded, and if line 29Ø is deleted, you may not find out if something goes wrong. By leaving 29Ø in, you can be sure that "No news is good news."

CHAPTER 2
HI-RES CHARACTER GENERATOR

The Hi-Res Character Generator (HRCG) is a set of assembly-language subroutines for displaying text on the high-resolution graphics screen. It serves the same function as the Apple's hardware character generator (described in the section The Video Generator in the Apple II Reference Manual). It turns a series of ASCII character codes into a bit-by-bit map of the Apple screen: each one-byte ASCII code produces a 7-wide-by-8-high dot image on the high-resolution screen, covering an area equal to two low-resolution blocks stacked vertically. Each dot will be on (white) or off (black), and the dots in each line of a character-image can be shifted sideways by half a dot to produce smoother contours. The Hi-Res Character Generator can produce the same character set as the hardware character generator, and many others besides: foreign alphabets and decorative alphabets, both with upper- and lowercase, and graphic alphabets for animation, which carve up a large image into 7x8 blocks which can be displayed anywhere on the screen.

With its default settings, the HRCG behaves much like the standard text display. The text window fills the screen, the screen scrolls, and Applesoft's VTAB and HTAB can be used to control the print position.

Beyond its ability to display upper- and lowercase text, the Hi-Res Character Generator provides several additional display capabilities:

* Intermixing of text and high-resolution graphics
* Normal and Inverse video
* Alternate character sets for user-defined characters
* The ability to display graphic character sets for animation
* The ability to write text over an existing background
* Scrolling or text wrap-around within the text window
* Automatic downshifting of alphabetic characters for displaying lowercase text

The Hi-Res Character Generator is stored under the filename HRCG on the Applesoft Tool Kit diskette. The diskette also includes the program LOADHRCG, which loads and initializes the HRCG program, as well as a number of character-set files. These include ASCII.SET, a standard ASCII character set; GREEK.SET, a Greek character set; many decorative alphabets; and several graphic character sets for animation. The diskette also contains the programs RIBBIT, SKYLAB, and MAXWELL, which show how the HRCG is used for animation.

LOADING CHARACTER SETS

To see how the HRCG works, we will use the program LOADHRCG to load and initialize it. Normally the HRCG would be loaded by a "host" program, like RIBBIT, which will provide the HRCG with text to display. How to use the HRCG under program control will be explained later in this chapter, but for now it will be simpler to control it directly by keyboard commands.

Before using the HRCG, we must load the desired character sets from diskette. To do this, first make sure you have DOS booted, then CATALOG the Applesoft Tool Kit diskette and write down the filenames of the character sets you want. Now reset HIMEM by typing

FP

then load the HRCG by typing

RUN LOADHRCG

(Note: It is always wise to use the FP command before loading the Character Generator, especially if you just used the Character Generator, then left the program. The Character Generator changes the value of HIMEM, and FP resets HIMEM to its default value. If you run LOADHRCG right after booting the system, FP is unnecessary.)

You will be asked how many character sets you wish to load. Answer by typing Ø if you will only use the standard set, or by typing a number from 1 to 9 if you want to load additional ones; then press RETURN. This will make the program allocate enough space in memory to hold your character sets. When the program is loaded, you will be asked for the name of the first alternate character set. Type the filename of this set, then press RETURN. Repeat until you have loaded all the character sets you want.

To see the alternate character sets, you can type

CTRL-A

followed by the number, n, from 1 to 9, of the alternate character set. Ignore the SYNTAX ERROR message, which results from the fact that the command to the HRCG is not a valid BASIC statement. You can suppress the error message by typing

CTRL-A n CTRL-X

(with no spaces). If you now LIST the program, LOADHRCG, it will come out in the new character set. To get back to the standard character set, type

CTRL-A Ø CTRL-X

(with no space). If you LIST the program, it will come out in the normal letters, but it will scroll differently: it will move up one bit at a time, rather than one byte; or one scan line, rather than one line of text. It moves at about the same rate, but in smaller jumps. This quirk, which derives from the bit-by-bit way in which the characters are produced, is a good way to recognize the Hi-Res Character Generator in operation. Each line of text is created one scan line at a time, rather than one character block at a time.

You can do this with any of the character sets you have loaded, to see what they look like. If you wish to change any of them, use ANIMATRIX, as described in its own chapter.

You may be amused by the results if you use one of the graphic character sets, say FROGS1.SET or SKY1.SET, to LIST the program. As always, typing
CTRL-A Ø CTRL-X
will get you back to the normal character set. If you press RESET, you will leave the HRCG entirely, and erase the character sets you have loaded. After pressing RESET, remember to use the FP command to reset HIMEM: if you RUN LOADHRCG repeatedly without resetting HIMEM, you will eventually run out of memory.

THE COMMANDS

The Hi-Res Character Generator's display format is controlled by PRINTing control sequences of one or two ASCII characters. The control sequences are described here, with a summary table at the end of the section. Most control sequences consist of a single ASCII control character, but some sequences require two control characters, and one requires a control character and a digit. In the descriptions below, control characters are separated by spaces for clarity. These spaces should not be typed. Default parameters are starred.

CTRL-A n — Select Character Set n

This control sequence selects character set n, where n is an ASCII digit. Character set Ø is the standard ASCII character set, comprising the 96 printing ASCII characters. You may also define up to nine alternate character sets, each with 96 additional symbols. Your character sets must reside in a contiguous block of memory. Before an alternate character set can be selected, its base address must be POKEd (low byte first) into the HRCG locations 7R and 8R ($Ø7R and $Ø8R). LOADHRCG takes care of this POKEing, so if you have run LOADHRCG, you can use the CTRL-A n command without further ado.

CTRL-N — Normal Video Display *

CTRL-I — Inverse Video Display

Normal video displays white characters on a black background. Inverse video displays black characters on a white background.

CTRL-O CTRL-P Print *

CTRL-O CTRL-O Overstrike

CTRL-O CTRL-C Complement

CTRL-O CTRL-T Transparent Overlay

CTRL-O CTRL-R Reverse Overlay

In Print mode, the printed character (white on black or black on white, depending on Normal/Inverse mode) replaces the current display. In Overstrike mode, the printed character (white or black, depending on Normal/Inverse) overlays the existing display. In Complement mode with Normal video, the character portion of the existing display is complemented; with Inverse video, the background is complemented. In Transparent mode, the character is overlaid on the secondary screen and the result transferred to the primary screen. In Reverse mode with Normal video, the character portion of the secondary screen is complemented and the result transferred to the primary screen; with Inverse video, the background is complemented and the result transferred.

CTRL-O CTRL-A Primary Screen is Hi-Res Page 1 *

CTRL-O CTRL-B Primary Screen is Hi-Res Page 2

The HRCG always writes on the primary screen. This parameter specifies which Hi-Res graphics page will be used as the primary screen. Note that changing the primary screen does not automatically change which Hi-Res graphics page is being displayed.

CTRL-O CTRL-D Display Primary Screen

This parameter sets the display to the primary screen.

CTRL-O CTRL-S Scroll Text Window *

CTRL-O CTRL-W Wrap Text Window

This parameter controls the Character Generator's action when it reaches the bottom of the text window. When scrolling is selected, the Character Generator moves the current display up one line and replaces the bottom line with blanks. When wraparound is selected, it simply moves the cursor to the top line of the text window.

CTRL-Q Home Cursor

Normally, CTRL-Q moves the cursor to the upper left corner of the text window. If the Character Generator is operating in Block Display mode, however, the cursor is moved to the upper left corner of the block.

CTRL-C Carriage Return

Normally, CTRL-C operates like a carriage return, moving the cursor to the left side of the text window and down one line. If the Character Generator is operating in Block Display mode, however, the cursor is moved to the left side of the Block and down one line.

CTRL-P Clear Page

CTRL-E Clear to End of Line

CTRL-F Clear to End of Screen

These control characters serve the same functions as the escape sequences ESC @, ESC E, and ESC F, or the Monitor commands CALL -936, CALL -868, and CALL -958.

CTRL-Y Set Text Window (full screen) *

CTRL-V Set Text Window (upper left)

CTRL-W Set Text Window (lower right)

These control characters are used to set the text window. CTRL-Y sets the text window to the full screen. With the window set to the full screen, CTRL-V sets the upper left corner of the text window to the current cursor position. CTRL-W sets the lower right corner of the text window to the current cursor position.

CTRL-K Caps Lock *

CTRL-L Lower Case Display

CTRL-S Shift

Normally, the Character Generator prints uppercase alphabetics in uppercase and lowercase alphabetics in lowercase. However, when the Lower Case Display mode is used, uppercase alphabetics will be printed in lowercase. This provides a convenient way for a program to print lowercase text without having to convert the data. CTRL-S prints the next character in uppercase, then puts the Character Generator in Lower Case Display mode.

CTRL-B Begin Block Display

CTRL-D Delimit Block Display

Entering Block Display mode causes the Character Generator to save the current cursor position and parameter settings. When a Block Display Delimiter (CTRL-D) is found, the original parameter settings are restored. Block Display mode is normally used in conjunction with an alternate character set to display pictures composed of a block of characters. For example, a character set might be designed to represent playing cards, with each card composed of a 2 X 2 character block. The ace of spaces might be represented by the following block of letters, each standing for one quarter of the image:

```
AS
DF
```

Although this card could be printed with two PRINT statements and appropriate VTABs and HTABs, Block Display mode could be used to represent it with the string

CTRL-B CTRL-A 1 A S CTRL-C D F CTRL-D

(The spaces are inserted for clarity, and are not part of the string.) A whole deck of such strings could be saved in an array CARD$ for use in the rest of the program.

CTRL-Z Set Default Parameters

CTRL-Z resets all of the Character Generator's parameters to their default values.

CTRL-O CTRL-Y Call User's Subroutine A

CTRL-O CTRL-Z Call User's Subroutine B

These control sequences cause the Character Generator to call a user-supplied assembly subroutine by performing a JSR. Before calling a user subroutine, the user must POKE the subroutine's address (low byte first) into the character generator at locations 1ØR and 11R ($ØAR and $ØBR) for subroutine A and locations 13R and 14R ($ØDR and $ØER) for subroutine B. The subroutine addresses default to an RTS.

CONTROL SEQUENCE SUMMARY

CHAR	FUNCTION	OPTION	
CTRL-A n	Select Character Set n	CTRL-A	Primary is Page 1 *
CTRL-B	Begin Block Display	CTRL-B	Primary is Page 2
CTRL-C	Carriage Return	CTRL-C	Complement
CTRL-D	Delimit Block Display	CTRL-D	Display Primary
CTRL-E	Clear to End of Line		
CTRL-F	Clear to End of Screen		
CTRL-G			
CTRL-H			
CTRL-I	Inverse Video		
CTRL-J			
CTRL-K	Caps Lock *		
CTRL-L	Lower Case		
CTRL-M			
CTRL-N	Normal Video *		
CTRL-O	Option Select →	CTRL-O	Overstrike
CTRL-P	Clear Page	CTRL-P	Print *
CTRL-Q	Home Cursor		
CTRL-R		CTRL-R	Reverse Overlay
CTRL-S	Shift	CTRL-S	Scroll *
CTRL-T		CTRL-T	Transparent Overlay
CTRL-U			
CTRL-V	Set Text Window (upper-left)		
CTRL-W	Set Text Window (lower-right)	CTRL-W	Wrap
CTRL-X			
CTRL-Y	Set Text Window (full screen) *	CTRL-Y	Call User Sub A
CTRL-Z	Restore Default Parameters	CTRL-Z	Call User Sub B

* Default Parameter

The default Character Set is ASCII.SET

DIFFERENCES FROM STANDARD TEXT DISPLAY

Although the Hi-Res Character Generator is intended to mirror the operation of the standard text display, there are several differences:

* The cursor is a flashing underscore character _ .

* The Applesoft command HOME and the monitor Call -936 will set the print position to the upper left corner of the text window, but they will not clear the screen; substitute CTRL-P .

* The Monitor commands CALL -958 and CALL -868 do not clear to end of screen and clear to end of line; substitute CTRL-F and CTRL-E .

* The Applesoft commands NORMAL, INVERSE, and FLASH do not control the Hi-Res video. Programs must write NORMAL video and use CTRL-N and CTRL-I to select normal and inverse video. Flashing video is not provided by the Character Generator.

* The HRCG uses the DOS input/output hooks, so it cannot be used with any other program that uses the hooks.

LEAVING THE HI-RES CHARACTER GENERATOR

To get out of the HRCG, press RESET, then type
FP
to reset HIMEM. Don't use the TEXT command: it will get you back to the text screen, but the HRCG will still be running: you won't have a cursor.

USING THE HRCG UNDER PROGRAM CONTROL

The Hi-Res Character Generator is intended to be used under the control of a "host" program, such as RIBBIT, although it can be run by itself. The HRCG program and the standard character set together require approximately 2.ØK bytes of memory. Each alternate character set will require an additional .75K bytes of memory. The program will run on a 32K system if the second hi-res graphics page is not used, and on a 48K system if the second page is used: on a 32K system, the second hi-res page is in the section of memory occupied by DOS, so writing to the second page would clobber DOS.

To make the HRCG as versatile as possible, it was written in relocatable form, so that the host program can put it into the most convenient place in memory. Before it can be used, HRCG must be relocated and initialized. The program is relocated by RLOAD, the relocating loader (described in its own chapter). Then it is initialized by a CALL to one of the two entry points. Since the Character Generator may be relocated to run at any address, entry points and other significant addresses within the Generator are given in relocatable form. The program LOADHRCG, which loads HRCG, should be studied if you want to write your own programs to use the Hi-Res

Character Generator. Absolute addresses are computed by adding the relocatable addresses to the relocation base returned from RLOAD. Relocatable addresses are denoted by the suffix R.

The program's main entry point is at the relocatable address ØR ($ØØR). When it is CALLed, the following initialization is performed:

1. The Character Generator is linked into the DOS input and output hooks.

2. All parameters are set to their default values.

3. The display is set to full screen hi-res graphics with the primary page displayed.

4. The screen is cleared and the title is displayed.

A second entry point, which performs only steps 1 and 2, is located at 3R ($Ø3R). This entry point is provided for applications that must not write on hi-res page 1. These applications must set the primary screen to page 2 before any data are written. For identification purposes, the Character Generator's version number is multiplied by 1Ø and stored at location 6R ($Ø6R).

CHAPTER 3
ANIMATRIX

Animatrix is an Applesoft program that lets you create and edit character sets for the Hi-Res Character Generator. Each of these character sets is a set of 7-by-8-dot images, which can be put anywhere on the Apple screen. Each character image comprises 8 rows of 7 dots: each row is represented by one byte, which comprises 7 bits for the individual dots and one dot for a row-shift.

Unlike the Hi-Res Character Generator, Animatrix is not intended to be used under the control of another program. It is used to create character sets before you run the HRCG.

STARTING UP

Make sure that DOS is booted and that the Applesoft Tool Kit diskette is in drive 1. Type
FP
to reset HIMEM, then type
RUN ANIMATRIX

When the program starts, it will print
CHARACTER SET TO EDIT?

If you wish to edit an existing character set, type in the name of the file containing the character set. If you wish to create a new character set, press RETURN. For now, respond by typing
ASCII.SET

The screen contains four areas of information, which will be blank when you start. Now type the letters of the alphabet, slowly, and they will appear on the screen, one by one. On the left side of the screen is a grid (green if you have color) containing an expanded image of the characters you typed. At the upper right is a box labeled 'Letters', showing the standard ASCII image of the characters you typed, as it would be printed by a dot-matrix printer. Below this is a box labeled 'Graphic', containing actual-size images of the characters you typed, as defined in the character set you just loaded. The images in the Letters box are for identification only: they will not change when you edit the images in the Graphic box and the expanded grid. Below the Letters box is the word <UPPER> or <lower>, which tells you whether the keyboard input is read as upper- or lowercase characters.

For example, if you are editing the ASCII.SET and you type
ABCDEFG
in uppercase mode, you will see the standard images of these letters in the 'Letters' box, and their graphic images in the Graphic box, exactly as they look when displayed, and expanded in the grid, in a more schematic form. You will notice some interesting differences

between the various forms. First compare the A in the Letters box to the A in the Graphic box: the first is house-shaped, and the second is more triangular. This is because Animatrix and the Hi-Res Character Generator can shift each of the horizontal lines making up a character, thus smoothing its contours. Here the second and fourth rows of dots are shifted one half-dot to the right. Now look at the expanded grid: the expanded A is more jagged-looking than the Graphic A, because while the Graphic box shows the half-dot shift graphically, the grid shows it schematically with a white square on the border between A and B. Showing the shift schematically makes character editing easier, because editing is all done by turning dots on and off, and because the white square makes it certain whether a row is shifted or not.

Now type

CTRL-Z

and you will be in lowercase mode, indicated by the <lower> below the Letters box. If you now type

abcdefg

you will see identifying images in the Letters box, and graphic images in the Graphic box and the expanded grid. If you wish to switch back to uppercase mode, type

CTRL-A

and keyboard input will be read as uppercase characters.

If you want to see the entire character set at once, typing

CTRL-D

gives you the character set in tabular form, with its name at the top, and five columns of characters. In each column the numbers on the left are the ASCII codes, the characters in the middle the standard ASCII characters, and the characters on the right the characters of the named character set, ASCII.SET. Only the numbers from 32 to 126 are used, as the other numbers are used for control characters. (Character 32 is the space.) This table is especially helpful when you are working with graphic character sets: if you want to edit a particular graphic character, but can't remember the standard character it corresponds to, just look in the table.

To restore the grid, press the space bar.

You can also get a summary of commands, by typing

CTRL-I

To restore the grid, press the space bar.

GETTING CHARACTERS ONTO THE SCREEN

To edit or create an image for a given character, first decide where you want it to go, then move the cursor in the Letters box to the square you want, by using the arrow keys and the RETURN key: the arrows will move the cursor left and right, and RETURN will move it to the beginning of the next line, just as on the normal Apple screen.

Now press the key that generates the character you want to define or change: for example, to edit the letter A, press the A key while in uppercase mode. The letter will appear as described above. When you type a character, its standard alphabetic image appears in the Letters box and its graphic image, if it is defined in this character set, appears in the Graphic box and on the grid. If you are defining a new character set, the Graphic box and the grid will both be empty, even when you have filled the Letters box. CTRL-A switches to uppercase input, and CTRL-Z switches to lowercase. The current case is always displayed below the Letters box.

The underscore, _ , backslash, \ , and left bracket, [, characters cannot normally be typed from the Apple keyboard, but can be input using the following control characters:

Underscore	-	CTRL-O
Backslash	\	CTRL-L
Left bracket	[	CTRL-K

The same control characters are used by the APA &Keys command.

EDITING CHARACTERS

Once you have entered the characters to be defined or changed, you can edit them using the paddles and the image on the grid. The paddles move a flashing cursor on the grid: paddle Ø controls horizontal motion; paddle 1 controls vertical motion. When the curser has been positioned over the desired dot in a character image, button Ø will turn the dot off and button 1 will turn the dot on. Each change will be shown both on the grid and in the Graphic box.

CTRL-T will toggle the shift bit of the row the grid cursor is in. When on, this bit shifts the row half a dot to the right, which changes the color of the dot if a color monitor is used. This shift can be used to smooth the contours of a letter. The shift will only be visible on the actual-size image at bottom right, not the expanded image on the expanded grid. On the grid, the shift will be indicated schematically by a white square on the right border of the letter image.

DUPLICATING CHARACTERS

If you wish two cells in the grid to have the same image, simply input the same character in the corresponding cells of the Letters box. From now on, until you change the characters in the Letters box, these cells, in each box, will behave the same. If one is undefined, the other will be undefined as well. If you define one, the other will be defined the same way. If you change one, the other will change the same way.

This feature is mainly useful in defining graphic character sets, which may repeat the same pattern in different parts of the image. If, for example, you define a character set consisting of several different running people, each several characters high, and the same character is used for the head of each, then you can change all the heads by changing one. Or one character can be used for a head turned toward the screen, and one for a profiled head, and each can be used with any of the runners. This shortcut can save time, and improve consistency.

If you wish to create several characters that are similar, differing by only a few dots, here's an easy way. Any undefined character can be made a duplicate of a defined one, so that both characters (in the Letters box) have the same images (in the Graphic box and the grid). To do this, pick a defined character (to be duplicated) and an undefined character (to duplicate the first). Let's say you have already defined O and P, and you wish to define Q and R. Input the first defined character, O, in the Letters box, put the cursor on it, then type
CTRL-S

You will be asked,

Same image for which character?

Now input the undefined character, Q. In the Letters box, the newly defined Q will replace the already defined O anywhere it occurs, and the cursor will move to the next space. In the Graphic box, all squares corresponding to the newly defined character Q will display the new image, which will be like that for O. Now you can repeat this process to make R like P. You can now edit the images for Q and R, to differentiate them from O and P. A duplicate image created with CTRL-S can be edited independently from its original: since they are linked to different characters, they will not change as one.

CTRL-S may only be used to define new characters: it will not change a character that has already been defined.

GRAPHIC CHARACTER SETS

To see what a graphic character set looks like, load FROGS1.SET and type in the following array of letters:

```
ABCDEQ
FGHIJKR
  LMNOP
!#  ABC
$%lmnoj
```

As you type in the letters, various creatures will appear on the screen.

LEAVING ANIMATRIX

To exit the program, press ESC. If you were editing an existing character set, you will be asked whether you wish to save the character set under its old filename. If you reply
Y
the old character set will be replaced by the new one. If you reply
N
or if you were creating a new character set, you will be asked for a new filename. When you type the new name and press RETURN, the new character set will be saved under the new name, and the old character set will be left undisturbed. If you press RETURN without typing a filename, the new character set will be erased as you leave the program.

At this point, you still have a chance to back out. After any of these options, you will then be asked whether you really want to quit. If you answer Y, you'll leave Animatrix; if you answer anything else, you'll return to the program, your current character set still in memory, and your old character sets unchanged.

If you leave Animatrix, you will still be in the Hi-Res Character Generator, and the character set you defined will be alternate character set 1.

To edit another character set, just type
FP
then RUN ANIMATRIX again.

SUMMARY OF ANIMATRIX COMMANDS

STRING INPUT COMMANDS

Left arrow	move cursor left
Right arrow	move cursor right
RETURN	go to next line
CTRL-A	shift to upper case
CTRL-K	left bracket, [
CTRL-L	backslash, \
CTRL-O	underscore, _
CTRL-Z	shift to lower case

CHARACTER EDITING COMMANDS

Paddle Ø	move grid cursor horizontally
Paddle 1	move grid cursor vertically
Button Ø	turn bit off
Button 1	turn bit on
CTRL-T	toggle high bit for half-dot shift
CTRL-S	same character image

OTHER COMMANDS

ESC	save character set and quit
CTRL-D	display entire character set
CTRL-I	display summary of instructions

CHAPTER 4

THE RELOCATING LOADER

The Relocating Loader lets you load Relocatable assembly-language modules (R files) from an Applesoft program, so that the program can run on Apples of different memory sizes. The Relocating Loader, which is made up of two programs, RLOAD and RBOOT, loads R files just below HIMEM, reducing HIMEM by the length of the modules loaded plus one byte (by length+1).

RESTRICTIONS

If your program uses string variables, use RBOOT and RLOAD before allocating or DIMensioning any strings, because RLOAD will be "hidden" between the memory locations containing your program lines and the locations containing your strings: if you allocate strings and then use RLOAD, RLOAD may land on top of your strings.

RBOOT

RBOOT is a small program that is BLOADed into addresses $2Ø8 through $3CF and invoked with a CALL 52Ø. Note that the usual D$ for the DOS BLOAD should be replaced with a CHR$(4), because you cannot allocate the string D$. RBOOT loads RLOAD above the end of APPLESOFT's variable table and sets the USR function to point to the entry point of RLOAD.

RLOAD

Once RLOAD has been set up, it can be used to load Relocatable modules. When accessed via the USR(Ø) function, RLOAD returns the starting address of the module. An "ON ERROR" return indicates that a problem was encountered during the attempted load. RBOOT and RLOAD both assume that the ON ERR statement is executed before calling them. The following example shows the syntax for RBOOT and RLOAD:

```
1ØØ  ADRS=Ø
11Ø  PRINT CHR$(4);"BLOAD RBOOT":CALL 52Ø
12Ø  ADRS=USR(Ø),"MYMODULE, S6, D1"
```

The slot and drive parameters are optional; the filename "MYMODULE" must refer to a Relocatable file.

The program LOADHRCG exemplifies the use of the Relocating Loader.

INDEX

A

absolute addresses 20
alternate character sets 20
Animatrix 1, 15, 23
  Commands, Summary of 28
  leaving 27
Apple II 1
Apple II Plus 1
Applesoft Firmware Card 1
Applesoft II 1
Applesoft Programmer's
    Assistant (APA) 1, 3
  commands 4
  parameters 4
ASCII 13, 24
Assembler/Editor 1
assembly language, 6502 1
&Auto 4, 10

B

backslash 9, 25
Begin Block Display 18
bit map 13
Block Display mode 18
bracket, left 9, 25
button inputs 25

C

Call User's Subroutine A 19
Call User's Subroutine B 19
CALL -868 17
CALL -868 20
CALL -936 17
CALL -936 20
CALL -958 17
CALL -958 20
Caps Lock 18
Carriage Return 17
character editing commands 28
characters, defining similar 26
characters, duplicating 25
characters, editing 25
characters, input of 24
Clear Page 17
Clear to End of Line 17
Clear to End of Screen 17
commands, APA 4-10
Command Summary, Animatrix 28
Complement mode 16
Control Sequence Summary, HRCG 19
CTRL-A 14, 15, 24, 25
CTRL-A n 15
CTRL-B 18
CTRL-C 17
CTRL-D 18, 24, 29
CTRL-E 17, 20
CTRL-F 17, 20
CTRL-I 15, 20, 24
CTRL-K 9, 18, 25
CTRL-L 9, 18, 25
CTRL-N 15, 20
CTRL-O 9, 25
CTRL-O CTRL-A 16
CTRL-O CTRL-B 16
CTRL-O CTRL-C 16
CTRL-O CTRL-D 16
CTRL-O CTRL-O 16
CTRL-O CTRL-P 16
CTRL-O CTRL-R 16
CTRL-O CTRL-S 16
CTRL-O CTRL-T 16
CTRL-O CTRL-W 16
CTRL-O CTRL-Y 19
CTRL-O CTRL-Z 19
CTRL-P 17, 20
CTRL-Q 17
CTRL-S 10, 18, 26
CTRL-T 25
CTRL-V 17
CTRL-W 17
CTRL-X 4, 14
CTRL-Y 17
CTRL-Z 18, 24, 25
cursor, HRCG 20

D

Default Character Set, HRCG 19
Default Parameters, HRCG 19
Delimit Block Display 18
disk controller PROMs 1
Display Primary Screen 16
DOS 1, 3, 14, 23
DUPLICATE LINE NUMBER 8

E

ESC E 17
ESC F 17
ESC @ 17
expanded grid 23

F

<first> 6
FLASH 20
flashing underscore 20
flashing video 20
FP 14, 20

G

GOSUB 6
GOTO 6
Graphic box 23-25
graphic character sets 24, 26

H

hardware character generator 13
high-resolution screen 13
HIMEM 3, 5, 14, 20, 23, 29
Hi-Res Character Generator
(HRCG) 1, 13-21
commands 15-19
leaving 20
recognizing in operation 14
initialization 20
&Hold 7, 11
HOME 20
Home Cursor 17
horses, king's 8
host program 14
HTAB 13

I

<inc> 4
input/output hooks 20
Integer BASIC 1
INTERLEAVED OR DUPLICATE
LINE NUMBER 6
INVERSE 20
inverse video 20
Inverse Video Display 15

J

K

&Keys 9
king 8

L

Language Card 1
Language System 1
<last> 6
left bracket 9, 25
&Length 9
Letters box 23, 24
LOADAPA 3, 10, 11, 29
Lower Case Display 18

M

&MAnual 5
MAXWELL 1, 13
memory 1
men, king's 8
&Merge 7, 11
Monitor commands 17, 20

N

n, number of alternate
character sets 14
NORMAL 20
normal video 20
Normal Video Display 15
&Noshow 9

O

ON ERROR 29
ONERR 6
Overstrike mode 16

P

paddle inputs 25
parameters, APA 4
Primary Screen is Hi-Res Page 1 16
Primary Screen is Hi-Res Page 2 16

Print mode 16
P5A 1
P6A 1

Q

R

R files 29
RBOOT 29
relocatable addresses 20
Relocatable modules 29
Relocating Loader 1, 11, 29
REMarks 9
&Renumber 5-7
  parameters, default values for 6
Reverse Overlay mode 16
RIBBIT 1, 10, 13
RLOAD 29

S

Scroll Text Window 16
Select Character Set n 15
Set Default Paramaters 18
Set Text Window (full screen) 17
Set Text Window (lower right) 17
Set Text Window (upper left) 17
Shift 18
&Show 9
SKYLAB 1, 13
<start> 4
string input commands 28
syntax, APA commands 3-4

T

text window 13
Transparent Overlay mode 16

U

underscore 9, 25

V

variables, Applesoft 10
Video Generator 13
VTAB 13

W

Wrap Text Window 16

X

&Xref 10

Y

Z

Addendum to the
Applesoft Tool Kit Manual

The information here supplements the information in the Applesoft Tool Kit Manual. When specific pages in that manual are relevant to the following points, the manual page number(s) are mentioned so that you can refer to the manual as you read this. You may wish to note some of these points in your manual.

page

5 The Note on this page warns you not to change the input/output hooks while the APA is in memory (by typing a PR# or IN # command or RUNning a program which does). Actually, there are times when the I/O hooks can be changed while the APA is loaded, depending upon which of the APA commands you are using. The only situations you MUST avoid are:

* Using a PR# command with either of the &Show and &Noshow commands.

* Performing an IN# command in conjunction with any of the &Auto, &Keys and &MAnual commands.

Remember that you must not change HIMEM or use the FP command while the APA in in memory.

1Ø The APA's &Xref command lists each variable followed by the numbers of the lines in which it appears. Unlike Applesoft's TRACE command, &Xref lists each line number only once for each variable, even if there are multiple statements in a single line which contain the same variable.

11 To leave the APA, press RESET. Then type
FP
to reset HIMEM.

11 It may be possible to have the APA in memory at the same time as another routine that uses the ampersand vector. To use such a routine without losing the APA, you must first perform a
CALL {address}
where {address} is the memory location of the beginning of that routine. Then the ampersand vector should be reset by the routine. The routine can then be accessed with the & function.

(continued on other side)

To return to the APA without losing the other routine, follow the same procedure for the APA. The beginning address of the APA would have to have been determined beforehand, immediately after you loaded the APA. To do this, you type
PRINT ADRS
right after loading the APA. The number you'd see then would be the correct beginning address of the APA for that session. Don't forget to determine this right after loading the APA if you plan to be using another routine which resets the ampersand vector.

13 The Hi-Res Character Generator can produce a number of character sets in both upper and lower case. The character sets Shadow.Set and Graphic.Set, however, do not have pre-defined lower case characters in the HRCG.

13 While the HRCG is in memory, do not use the FP command, change HIMEM, change the input/output hooks (with a PR# or IN # command) or RUN a program which does any of these things.

A final point about both the APA and the HRCG:

Before switching to Integer BASIC, you must leave the APA or the HRCG (if either is in memory). Press RESET, then type
INT

10260 Bandley Drive
Cupertino, California 95014
(408) 996-1010

030-0113-00